

REMARKS

Claims 1, 7, 9 and 16 have been amended, while claims 2-6 and 17-20 have been canceled without prejudice. Applicant respectfully traverses the Office's rejections and, in view of the foregoing amendments and the following remarks, respectfully requests that the Office issue a Notice of Allowance.

§§ 102 AND 103 REJECTIONS

Claims 1-3, 6, 8, 9-10, 12, and 16-18 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Altiris RapidInstall, version 3.0, "Release Notes" (hereinafter, "RapidInstall").

Claims 7 and 11 stand rejected under 35 U.S.C. § 102(b) as anticipated by or, in the alternative, under 35 U.S.C. § 103(a) as being obvious over RapidInstall.

Claims 4 and 19 stand rejected under 35 U.S.C. § 103(a) as being obvious over RapidInstall in view of Dykhuis, R., "Beefing up DOS with 4DOS" (hereinafter, "Dykhuis").

Claims 5 and 20 stand rejected under 35 U.S.C. §103(a) as being obvious over RapidInstall in view of Yager, T., "Taking Command of Windows NT" (hereinafter, "Yager").

Claims 13-15 stand rejected under 35 U.S.C. §103(a) as being obvious over RapidInstall in view of U.S. Patent No. 7,103,590 to Murthy et al. (hereinafter, "Murthy").

THE REJECTIONS

Applicant respectfully submits that the Office has not established that the RapidInstall reference anticipates or renders obvious Applicant's pending claims. Nevertheless, Applicant has amended the independent claims as shown above. Applicant respectfully submits that these amendments obviate the grounds for the rejections, as discussed in detail below, and respectfully requests that the Office issue a Notice of Allowance.

The discussion begins with a section entitled "Applicant's Disclosure", which describes salient aspects of the present Application. A section entitled "The RapidInstall Reference" follows and explains the RapidInstall reference. Finally, a section entitled "The Claims" presents Applicant's reasoning as to why the cited references fail to anticipate or render obvious Applicant's amended claims.

Applicant's Disclosure

Applicant herein provides a brief description of the current Application. Applicant notes that while this description is provided to assist the reader in appreciating the subject matter described in Applicant's specification, it is not intended to imply or impose specific limitations with regard to Applicant's claims.

Perhaps a good place to start to gain an appreciation of the claimed subject matter is in Applicant's "Background" section. There, the "Background" section describes that in a command line environment, a command line interface allows a user to directly perform a task by entering in a command. For example, a command line interface may be invoked that provides a window that displays a prompt (e.g., "C:\>"). A user may type in any of numerous commands, such as

“dir”, at the prompt to perform the command. *Applicant’s Specification*, Background.

With this setting in mind, the present Application provides extended functionality to command line instructions in a manner that is significantly different from the traditional mechanisms for extending functionality. For example, in traditional mechanisms, each command that desires the extended functionality must ***incorporate the corresponding code into the command itself***. See, e.g., *RapidInstall infra*. The command itself must then parse the command string to determine whether a switch (e.g., verbose, whatif) was provided and execute the extended functionality accordingly. ***In contrast***, mechanisms described within the present Application allow users to specify an argument within the command string in order to execute the extended functionality for a particular cmdlet, as long as the cmdlet incorporates a hook to the extended functionality. That is, an operating environment itself provides the extended functionality—and not merely the command itself. Thus, the present mechanisms *minimize the amount of code system administrators need to write*. In addition, the *extended functionality is implemented in a uniform manner*. *Applicant’s Specification*, p. 74, lines 7-20. (emphasis added).

The RapidInstall Reference

RapidInstall, meanwhile, describes a traditional mechanism as discussed above. Functionally, the RapidInstall application captures changes that occur on a personal computer when the user installs an application or makes configuration changes. RapidInstall then captures and builds these changes into a compressed,

self-contained, and self-extracting *executable (.exe) file called a RapidInstall Package (or “RIP”)*. This RIP file can then be loaded onto client personal computers to make for easier installation of the new applications and configuration changes.

RapidInstall also discusses various command line switches (e.g., “-r”, “-cu”, “-si”) that can be executed from a command line. However, importantly, the RapidInstall program itself incorporates the code that handles these switches and functionality. One specific command line switch provided by RapidInstall is -si, which allows for a “Simulated Install”. Here, a user may simulate the installation of a built RIP file to determine if the user may experience any potential problems upon actual installation of the RIP file. Again, the user may use the “-si command line” to view this simulation. When the user runs the simulation via the -si command line, the RIP runs without actually making changes to the system, and a log flags any possible deployment problems. *RapidInstall 3.0 Release Notes*, p. 1. Again, the -si command *itself* provides this simulation functionality.

The Claims

Claim 1 recites in a command line operating environment, a computer-executable method comprising (emphasis added):

- executing each command on a command line in a first execution mode or in an alternate execution mode, wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode, *the alternate execution mode being provided by the command line operating environment such that the command being executed on the command line does not incorporate code for extending functionality to execute the command in the alternate execution mode;*

- wherein the instruction to execute in the alternate execution mode comprises a switch, and wherein executing the command in the alternate execution mode further occurs when the command line includes the switch indicating the alternate execution mode, and wherein the alternate execution mode visually displays simulated results of executing the command.

In making out a rejection of this claim, the Office states that RapidInstall anticipates and cites to the following portion of the reference: “page 1, section ‘Simulated Install –si command line”. *Office Action mailed 01/25/2008*, p. 2-3. Applicant respectfully traverses the rejection. Nevertheless, for the sole purpose of expediting allowance and without conceding the propriety of the rejection, Applicant has amended this claim. Applicant respectfully submits that RapidInstall, as well as the remaining cited references, fail to teach or suggest [an] “alternate execution mode *being provided by the command line operating environment such that the command being executed on the command line does not incorporate code for extending functionality to execute the command in the alternate execution mode*”, as recited in Applicant’s amended claim. (emphasis added).

To understand the importance of this claim element, Applicant first directs the Office’s attention to the section entitled “Applicant’s Disclosure” above. In brief, this section states that traditional mechanisms for providing extended functionality to command line instructions require that each command that desires the extended functionality ***actually incorporate the code into the command itself***, as opposed to “being provided by the command line operating environment *such that the command being executed on the command line does not incorporate code for extending functionality to execute the command in the alternate execution*

mode". Applicant respectfully submits that the RapidInstall reference merely incorporates its functionality into the code itself in accordance with traditional mechanisms.

Specifically, the RapidInstall reference provides for a "Simulated Install" of an RIP file (i.e., *an executable file*). This simulation allows a user to simulate the installation of a built RIP file in order to determine if the user will experience any potential problems upon actual installation of the RIP file. To do so, the user may use the "-si command line". *RapidInstall 3.0 Release Notes*, p. 1. However, RapidInstall does not allow the user to simulate any command or file *other than the RIP file*.

As such, Applicant respectfully submits that RapidInstall incorporates the simulate command into the RapidInstall command itself. That is, the -si command *itself* provides for the simulation, in contrast with a simulation "*being provided by the command line operating environment such that the command being executed on the command line does not incorporate code for extending functionality to execute the command in the alternate execution mode*", as recited in Applicant's claim 1. (emphasis added).

To further highlight this vital distinction, imagine that a RapidInstall user wishes to simulate the installation of an RIP file. This user would first enter the -si command into a command line operating environment. *Based on the command itself*, RapidInstall would then simulate the results of running the RIP file. Importantly, if the user wished to view the simulation of a *second* installation of a file *unrelated* to the RIP file, the traditional command line operating environment would be unable to so provide the user. Furthermore, because the -si command of

RapidInstall only relates to simulating the RIP file, RapidInstall would similarly not help the user in viewing simulated results of the second installation. As discussed at length above, this failing of RapidInstall is due to the fact that the simulation functionality is provided by the *–si command itself—and is not “being provided by the command line operating environment such that the command being executed on the command line does not incorporate code for extending functionality to execute the command in the alternate execution mode”*, as recited in Applicant’s claim. (emphasis added).

Applicant therefore respectfully submits that the RapidInstall reference fails to anticipate or render obvious Applicant’s claim 1, as RapidInstall cannot be shown to disclose “wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode, *the alternate execution mode being provided by the command line operating environment such that the command being executed on the command line does not incorporate code for extending functionality to execute the command in the alternate execution mode*”, as recited in Applicant’s claim. (emphasis added).

During the afore-mentioned interview, Applicant understood the office to tentatively agree that this language appears to overcome the RapidInstall reference. However, the Office also stated that further consideration of the RapidInstall reference and an updated search would be required before the Office made a final determination. Applicant thanks the Office for the Office’s reconsideration of this matter in view of Applicant’s amendments to claim 1.

For at least the reasons discussed above, Applicant respectfully submits that claim 1 stands allowable.

Claims 6-8 depend from claim 1 and, as such, the remarks made above in regards to claim 1 apply equally to these claims. These claims also stand allowable for their own recited features which, in combination with those recited in claim 1, are not shown to be disclosed or suggested in the reference of record.

Claim 9 recites at least one computer-readable medium having computer-executable instructions for performing a method, comprising (emphasis added):

- receiving a command line operative by a command line operating environment, the command line including a command that directs the performance of a task;
- determining by the command line operating environment if a parameter is present on the command line that is associated with a simulation mode;
- if the parameter is present, simulating the performance of the task by the command line operating environment such that the command being executed on the command line does not incorporate code for simulating the performance of the task; and
- visually displaying the results of the simulation by the command line operating environment.

In making out a rejection of this claim, the Office states that RapidInstall anticipates. Applicant respectfully disagrees. Nevertheless, for the sole purpose of advancing prosecution and without conceding the propriety of the Office's rejections, Applicant has amended this claim in a manner similar to the amendment of claim 1. Therefore, Applicant respectfully submits that this claim stands allowable for at least reasons similar to those discussed above in regards to claim 1. For example, Applicant respectfully submits that RapidInstall at least

fails to disclose or suggest “simulating the performance of the task by the command line operating environment *such that the command being executed on the command line does not incorporate code for simulating the performance of the task*”, as recited in Applicant’s claim. (emphasis added).

Claims 10-15 depend from claim 9 and, as such, the remarks made above in regards to claim 9 apply equally to these claims. These claims also stand allowable for their own recited features which, in combination with those recited in claim 9, are not shown to be disclosed or suggested in the reference of record.

Claim 16 recites a system that provides a command line operating environment, the system comprising (emphasis added):

- a processor; and
- a memory, the memory being allocated for a plurality of computer-executable instructions which are loaded into the memory for execution by the processor, the computer-executable instructions performing a method comprising:
- executing each command entered on a command line, wherein if the command includes an instruction to execute the command using extended functionality provided by the command line operating environment, executing the command using the extended functionality;
- *wherein the command line operating environment provides the extended functionality such that the command being executed on the command line does not incorporate code for extending functionality;* and
- wherein the extended functionality comprises visually displaying simulated results of executing the command.

In making out a rejection of this claim, the Office states that RapidInstall anticipates. Applicant respectfully disagrees. Nevertheless, for the sole purpose

of advancing prosecution and without conceding the propriety of the Office's rejections, Applicant has amended this claim in a manner similar to the amendment of claim 1. Therefore, Applicant respectfully submits that this claim stands allowable for at least reasons similar to those discussed above in regards to claim 1. For example, Applicant respectfully submits that RapidInstall at least fails to disclose or suggest "*wherein the command line operating environment provides the extended functionality such that the command being executed on the command line does not incorporate code for extending functionality*", as recited in Applicant's claim. (emphasis added).

Conclusion

For at least the foregoing reasons, Applicant respectfully submits claims 1 and 7-16 are in condition for allowance. Applicant respectfully requests reconsideration and withdrawal of the rejections and an early notice of allowance. If any issue remains unresolved that would prevent allowance of this case, Applicant respectfully requests the Office to contact the undersigned attorney to resolve the issue

Respectfully Submitted,

Dated: 2008/05/06

By: /Robert G. Hartman 58,970/

Robert G. Hartman
Lee & Hayes, PLLC
Reg. No. 58,970
(509) 324-9256 ext. 265